

清华大学

TSINGHUA UNIVERSITY

# 《智能信息处理》 第五组项目报告

Project Report of Team 5



项目：Kaggle-基于市场及新闻数据的股票预测

小组成员：徐哲、张凌霄、许卓群、路万通、于苗苗

## 目录

1 项目背景 .....	3
1.1 比赛概述.....	3
1.2 关键难点.....	4
2 探索性数据分析（EDA） .....	5
2.1 数据的来源.....	5
2.2 数据预分析.....	6
2.3 评价指标.....	12
3 神经网络模型.....	12
3.1 Market 数据预处理.....	12
3.1.1 分类变量预处理.....	13
3.1.2 数值型变量预处理.....	14
3.2 网络结构.....	16
3.3 训练与预测过程.....	19
3.4 排名情况.....	19
4 Boosting 方法-LightGBM 模型 .....	20
4.1 Market 数据特征工程.....	20
4.2 News 数据选取 .....	22
4.3 特征融合 .....	23
4.4 LightGMB 模型 .....	24
4.4.1 模型介绍 .....	24
4.4.2 参数选择 .....	25
4.5 排名情况.....	26
5 团队分工与最终成绩 .....	26
6 个人感想 .....	28

# 1 项目背景

## 1.1 比赛概述

**原始题目：** Can we use the content of news analytics to predict stock price performance?

The ubiquity of data today enables investors at any scale to make better investment decisions. The challenge is ingesting and interpreting the data to determine which data is useful, finding the signal in this sea of information. Two Sigma is passionate about this challenge and is excited to share it with the Kaggle community.

As a scientifically driven investment manager, Two Sigma has been applying technology and data science to financial forecasts for over 17 years. Their pioneering advances in big data, AI, and machine learning have pushed the investment industry forward. Now, they're eager to engage with Kagglers in this continuing pursuit of innovation.

By analyzing news data to predict stock prices, Kagglers have a unique opportunity to advance the state of research in understanding the predictive power of the news. This power, if harnessed, could help predict financial outcomes and generate significant economic impact all over the world.

Data for this competition comes from the following sources:

- Market data provided by Intrinio.
- News data provided by Thomson Reuters. Copyright ©, Thomson Reuters, 2017.

这次题目主题是使用新闻数据和股票历史数据来预测未来股票数据的涨跌。提出本次比赛题目的公司是美国纽约著名的对冲基金公司 TWO SIGMA，这是一家使用各种技术，包括人工智能，机器学习和分布式计算来进行交易策略分析的公司。

股市是一国经济的晴雨表，然而股市受政策、新闻、舆论的影响非常大，容易波动剧烈。因此对股市进行研究很有必要。随着互联网新媒体的发展，人们越来越倾向于通过互联网平台来交流信息，实时股评中包含丰富的金融信息，体现投资者的情绪变化，因此对股市的研究可以考虑从股评入手进行挖掘分析。文本

挖掘、机器学习、时间序列模型等技术兴起使得股评挖掘成为了可能。

在 2001 年 L.Cao 和 Francis E.H. Tay 等人阐述使用支持向量机模型 (SVM) 对标准普尔 500 指数进行分析, 通过采用几种指标作为特征, 并建模预测 1993-1994 年股票价格的走势, 结果达到了一定的准确率, SVM 也被广泛应用于金融领域。Hassan M.R 等人在 2007 年提出混合了 HMM (马尔科夫模型), ANN (人工神经网络) 和 GA (遗传算法) 的模型, 来使用从雅虎财经网站上收集的数据预测股票的变化, 结果比一般的单一模型效果有了进一步的提升。Duan W 和 Yu Y 在 2013 年基于情感分析的方法研究了社交媒体和传统媒体对于企业价值的影响, 发现社交媒体的情绪变化与上市公司股票的关联性更强, 市场情绪对于股票的走势也有一定的影响, 这一方向也有一定的研究价值。

目前关于股票走势研究的文章比较多, 但是都有一定的局限性, 所以 TWO SIGMA 公司在 Kaggle 上的赛题主要是希望参赛选手能够在股票走势方面得到更高的准确率。

## 1.2 关键难点

该比赛主要存在以下难点:

- a) Two Sigma 举办的比赛历来十分注重隐私性, 只能在 Kaggle 的 Kernels 上编写程序及运行, 公司不支持原始数据本地下载。
- b) 网络: Kaggle 由于是国外网站, 响应速度较慢且不稳定, 经常在跑程序时需要 reconnect 的问题。

```
LGBMClassifier(boosting_type='dart', class_weight=None, colsample_bytree=1.0,
                importance_type='split', learning_rate=0.1, max_depth=-1,
                min_child_samples=212, min_child_weight=0.001, min_split_gain=0.0,
                n_estimators=500, n_jobs=4, num_leaves=2452, objective='binary',
                random_state=100, reg_alpha=0.0, reg_lambda=0.01, silent=True,
                subsample=1.0, subsample_for_bin=200000, subsample_freq=0)
[Reconnecting]
[Running]
Your kernel is now running in the cloud. Here are some things you can do with it:
* Use the Play button or [SHIFT]+[ENTER] to execute the current line of your script (or whatever's highlighted).
* Enter some code at the bottom of this Console tab and press [ENTER].
[Reconnecting]
```

- c) Two Sigma 组委会特殊要求——不支持 GPU 计算的结果: 尽管可以用 Kaggle 上的 GPU 跑程序, 但是用 GPU 计算的结果不能提交评分 (故过

于复杂的模型不宜使用，先考虑快的方法如 **boosting** 相关的模型、传统的机器学习方法及采用简单单元的神经网络，放弃 CNN,RNN,LSTM 等运算复杂度太高的深度学习单元)。

- d) 组委会的限制给我们在**数据预处理、特征工程**提出了**更高的要求**，如 outliers 的处理、特征选取、基于现有特征计算出更好的新特征等。
- e) market 和 news 两者数据不平衡，若两者都想用上，需要做好特征选取及数据融合工作。

## 2 探索性数据分析 (EDA)

在正式的建立模型之前进行数据的预处理分析有利于在模型的建立时更好的处理特征。对数据的预处理是对数据更深度的挖掘，这对日后的模型计算会产生深远的影响。

### 2.1 数据的来源

本次比赛的数据来源主要来自于两个方面：

首先，第一方面是由 **Intrinio** 提供的 2007 年至今市场数据，其中包含金融场信息，如开盘价，收盘价，当前时间，资产的唯一 ID，与资产 ID 对应的名称，一个表示当天的工具是否包含在评分中的布尔值，成交量，当天的开盘价和收盘价和计算回报等。其中回报总是计算为未平仓（从一个交易日的开盘时间到另一个交易日的开盘时间）或接近收盘价（从一个交易日的收盘时间到另一个交易日的开盘时间）。回报是原始的，意味着数据不是根据任何基准进行调整，也不是市场残差。

第二部分是 2007 年至今新闻数据（资料来源：汤森路透），其中包含有关资产的新闻文章/警报信息，如文章详情，情绪和其他评论。

## 2.2 数据预分析

首先对新闻数据进行分析：

### (1) 新闻随时间变换关系

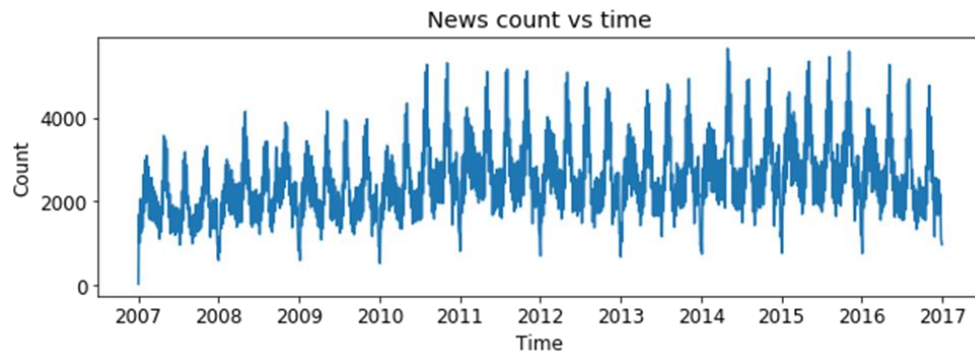


图 2.1 新闻数据随时间变化图

由以上图表可知，新闻数据量在每个季度呈周期性变化趋势，在每年年末（即圣诞节期间），新闻量达到最小值，且新闻数据量逐年上涨。

### (2) 新闻紧迫性分析

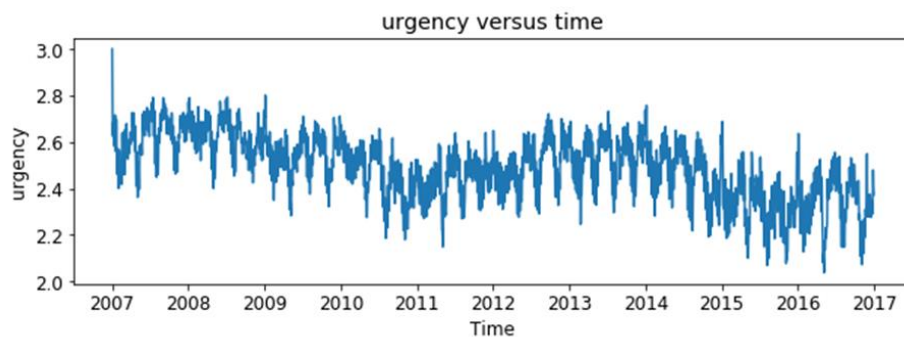


图 2.2 新闻紧迫性随时间变化图表

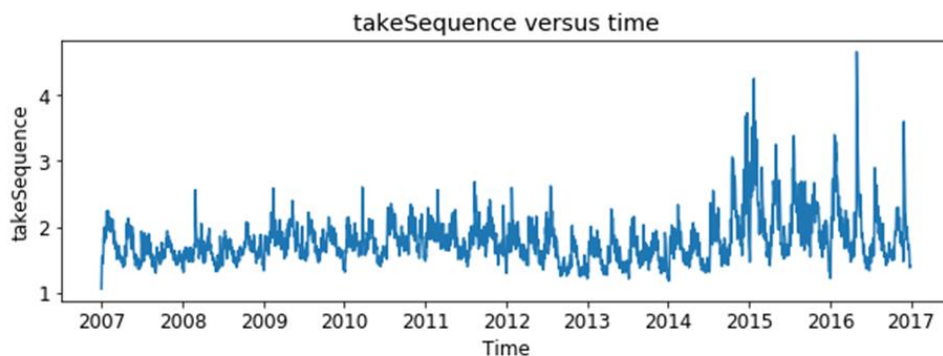


图 2.3 新闻项的获取序列号随时间变化图表

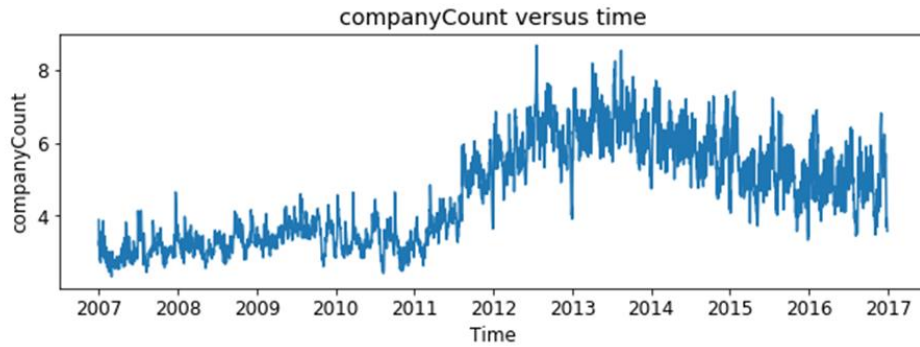


图 2.4 subject 字段的新闻项中明确列出的公司数随时间变化图表

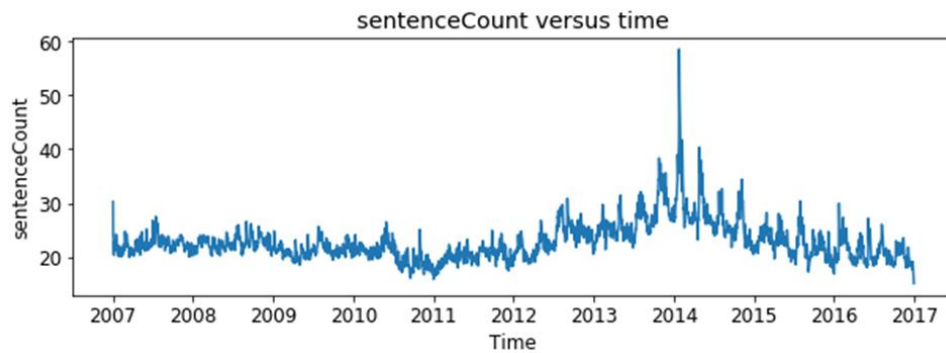


图 2.5 新闻条目中的句子总数随时间变化关系

由以上图表可知，新闻的紧迫性随时间变化降低，意味着新闻趋近于文章的趋势降低，趋近于警报的趋势升高。新闻中列出的公司数目和新闻中出现的句子总数总体呈上升趋势，这两者均在 2014 年达到峰值。

### (3) 资产与新闻的关系

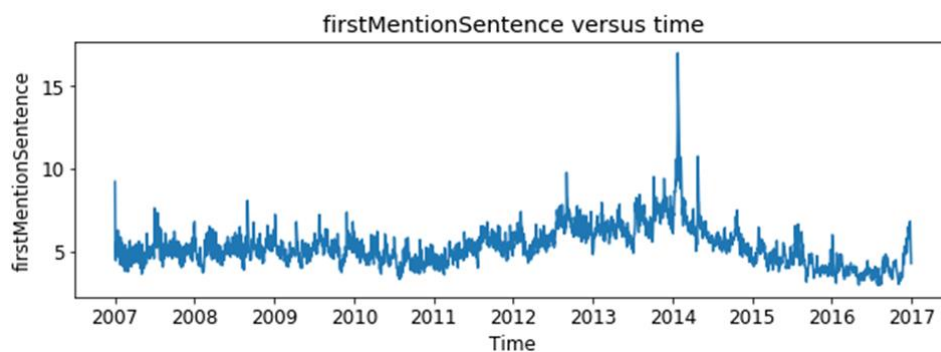


图 2.6 提到的评分资产的次序随时间变化图



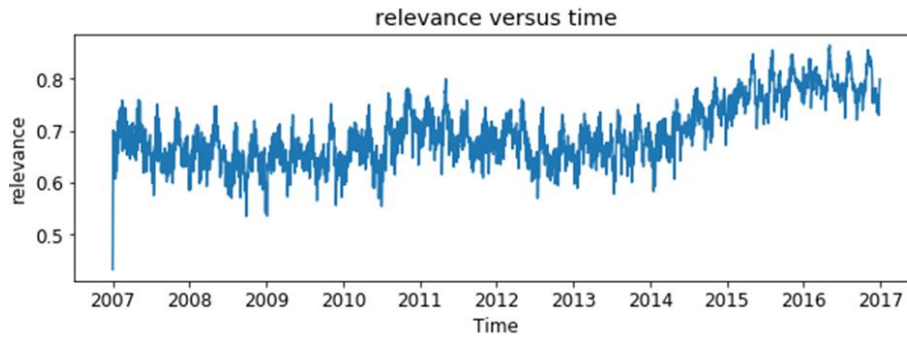


图 2.7 新闻项与资产的相关性随时间变化关系

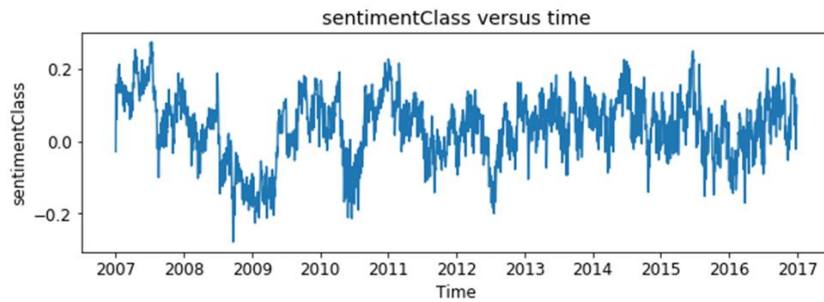


图 2.8 新闻项相对于资产的主要情绪类随时间变化图

由以上图表可知，新闻中提到的评分资产的次序整体呈下降趋势，在 2014 年达到最高峰值，说明在 2014 年的新闻内容中常出现与股票有关信息。新闻项与资产的相关性随着时间的变化增强，说明新闻与股票的关系越来越紧密。新闻项相对于资产的主要情绪呈周期性波动，其中在 2009 年明显呈消极趋势，这与 2009 年的经济危机相吻合。词法标记数随时间变化呈波动趋势，其中在 2014 年明显增多，这说明在 2014 年底的新闻中明显增多了对资产预测的信息。

#### (4) 资产的新颖性判断

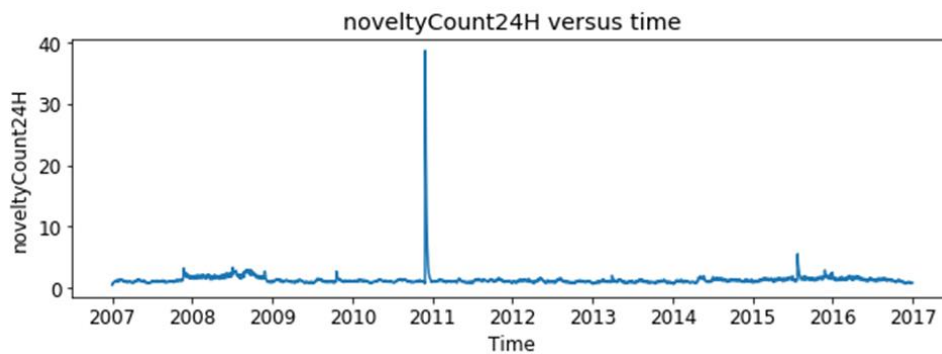


图 2.9 特定资产的新闻项目内容的 24 小时新颖性随时间变化关系



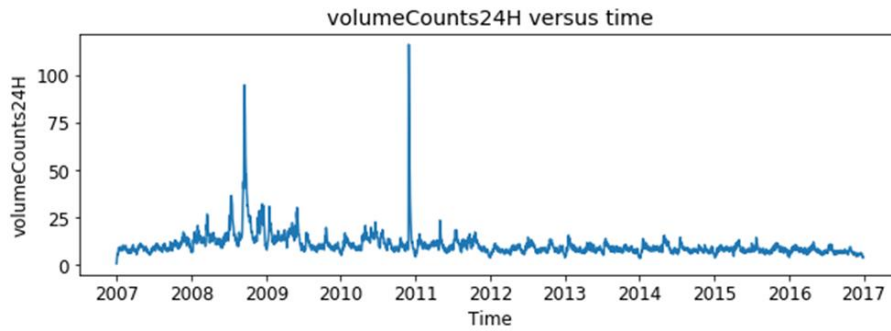


图 2.10 每个资产的 24 小时新闻量随时间变化关系

由以上图表可知，在 2011 年底，特定资产的新闻项目的 24 小时新颖性和每个资产的 24 小时新闻量明显增多，说明新闻的新颖性与新闻量成正比。

### (5) 延迟时间分布分析

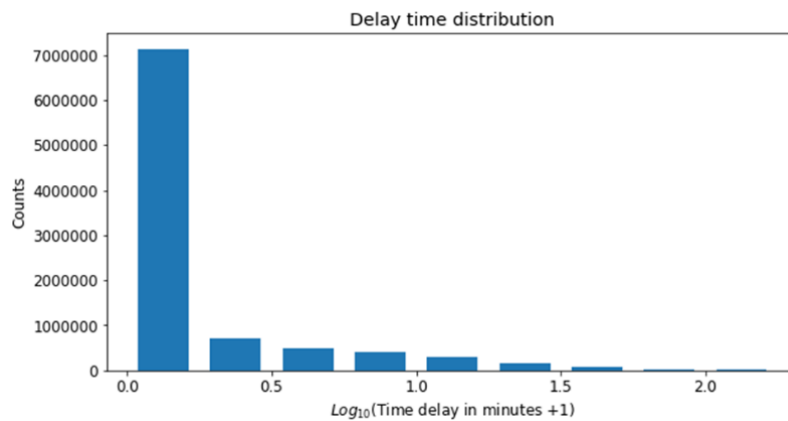


图 2.11 延迟时间分布随时间变化关系表

由以上图表可知，延迟时间分布的时间大部分都较短，说明新闻反映股票预测趋势具有延迟性。

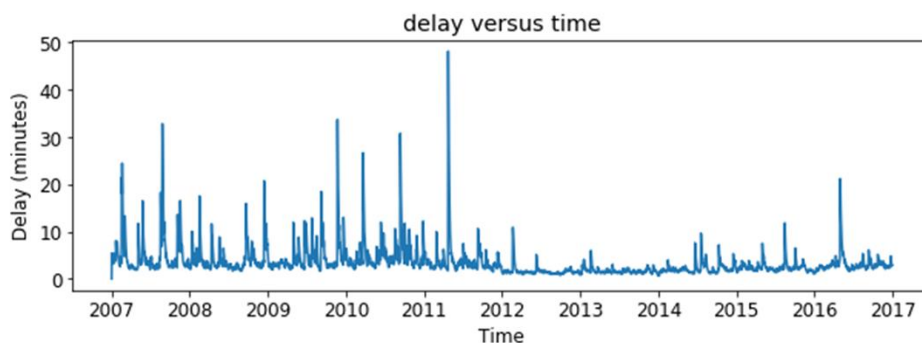


图 2.12 延迟时间随时间变化关系表

由上表可知，延迟时间随时间变化呈逐渐降低的趋势，说明新闻的时效性越来越强。

(6) 新闻信息的统计分布

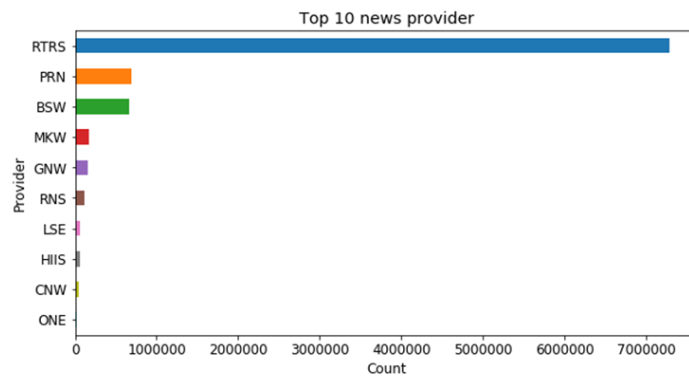


图 2.13 10 大新闻提供商

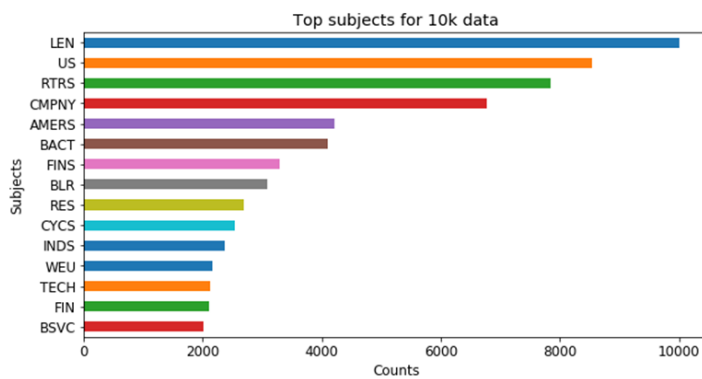


图 2.14 10 大新闻主题

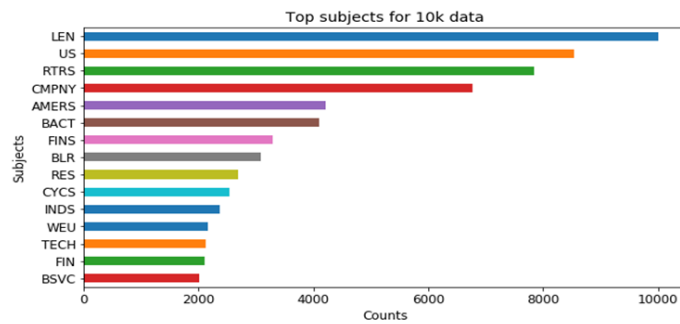


图 2.15 10 大受众

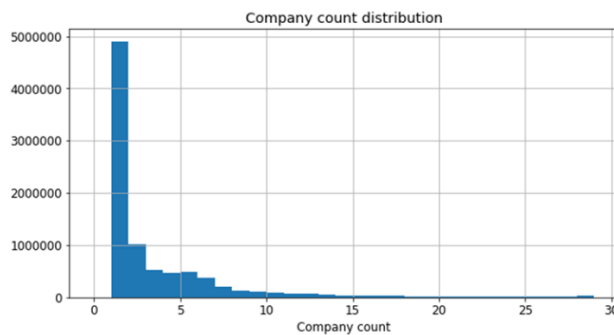


图 2.16 新闻中所提到公司数目分布

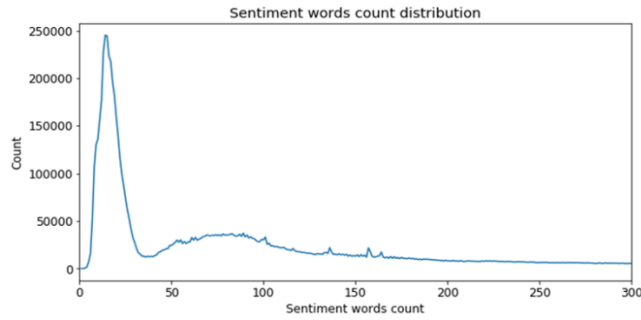


图 2.18 情感单词的数目分布表

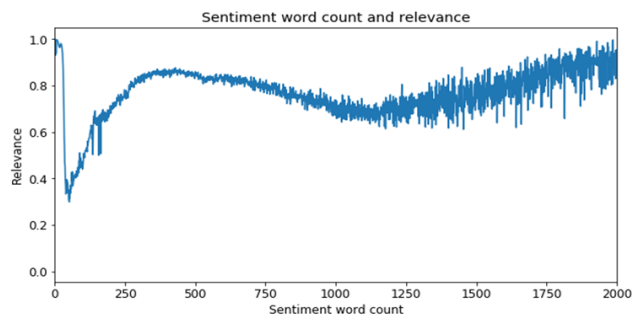


图 2.19 情感单词数目及其相关性

(7) 不同特征间关系分析

urgency	1	-0.57	0.2	0.15	0.51	0.3	-0.45	0.044	0.46	0.072	0.091	0.1	0.11	0.098	0.13	0.14	0.15
takeSequence	-0.57	1	0.0089	-0.077	-0.28	-0.16	0.22	-0.053	-0.27	0.045	0.017	-0.00079	-0.016	0.035	-0.025	-0.044	-0.055
companyCount	0.2	0.0089	1	0.25	0.53	0.58	-0.53	-0.046	-0.049	0.12	0.12	0.12	0.12	0.066	0.033	0.017	0.0023
marketCommentary	0.15	-0.077	0.25	1	0.28	0.29	-0.28	-0.059	-0.035	0.045	0.04	0.037	0.035	0.16	0.11	0.089	0.076
sentenceCount	0.51	-0.28	0.53	0.28	1	0.52	-0.43	-0.0071	0.49	0.13	0.14	0.14	0.15	0.062	0.059	0.054	0.047
firstMentionSentence	0.3	-0.16	0.58	0.29	0.52	1	-0.64	-0.047	-0.14	0.11	0.11	0.11	0.11	0.054	0.05	0.045	0.038
relevance	-0.45	0.22	-0.53	-0.28	-0.43	-0.64	1	0.1	0.013	0.014	0.0074	0.0041	0.0014	-0.14	-0.16	-0.17	-0.17
sentimentClass	0.044	-0.053	-0.046	-0.059	-0.0071	-0.047	0.1	1	0.058	0.011	0.018	0.023	0.028	-0.09	-0.088	-0.086	-0.082
sentimentWordCount	0.46	-0.27	-0.049	-0.035	0.49	-0.14	0.013	0.058	1	-0.063	-0.055	-0.051	-0.046	-0.025	-0.0055	0.0024	0.006
noveltyCount24H	0.072	0.045	0.12	0.045	0.13	0.11	0.014	0.011	-0.063	1	0.95	0.91	0.87	0.33	0.27	0.24	0.23
noveltyCount3D	0.091	0.017	0.12	0.04	0.14	0.11	0.0074	0.018	-0.055	0.95	1	0.97	0.94	0.34	0.31	0.29	0.27
noveltyCount5D	0.1	-0.00079	0.12	0.037	0.14	0.11	0.0041	0.023	-0.051	0.91	0.97	1	0.98	0.33	0.31	0.31	0.29
noveltyCount7D	0.11	-0.016	0.12	0.035	0.15	0.11	0.0014	0.028	-0.046	0.87	0.94	0.98	1	0.32	0.31	0.31	0.31
volumeCounts24H	0.098	0.035	0.066	0.16	0.062	0.054	-0.14	-0.09	-0.025	0.33	0.34	0.33	0.32	1	0.88	0.82	0.78
volumeCounts3D	0.13	-0.025	0.033	0.11	0.059	0.05	-0.16	-0.088	-0.0055	0.27	0.31	0.31	0.31	0.88	1	0.94	0.89
volumeCounts5D	0.14	-0.044	0.017	0.089	0.054	0.045	-0.17	-0.086	0.0024	0.24	0.29	0.31	0.31	0.82	0.94	1	0.97
volumeCounts7D	0.15	-0.055	0.0023	0.076	0.047	0.038	-0.17	-0.082	0.006	0.23	0.27	0.29	0.31	0.78	0.89	0.97	1

图 2.20 不同特征之间的关联度分布表

以上图表是不同特征之间的关联度分布表，当关联度接近于 1 时，说明两个特征之间强相关，若接近于 0，说明两个特征之间非强相关。通过不同特征之间的关系分布表，可以在以下的模型搭建中选择更合适的特征进行分析训练。

## 2.3 评价指标

在这次比赛中，我们需要预测一个有符号的置信度值  $\hat{y}_i \in [-1, 1]$ ，如果认为股票在未来十天内具有较大的正回报，则有正的置信度值（接近 1.0）。如果认为股票具有负回报，则可以为其指定一个较大的负置信度值（接近 -1.0）。如果不确定，则可以为其指定接近零的值。对于评估时间段内的每一天，我们计算：

$$x_t = \sum_i \hat{y}_i r_{it} u_{it}$$

其中  $r_{it}$  是工具  $i$  的第  $t$  天市场调整后的领先回报，而  $u_{it}$  是 0/1 布尔变量，控制特定资产是否包含在特定日期的评分中。然后，提交分数计算为平均值除以每日  $x_t$  值的标准差：

$$score = \frac{\bar{x}_t}{\sigma(x_t)}$$

## 3 神经网络模型

### 3.1 Market 数据预处理

在工程实践中，我们得到的数据会存在有缺失值、重复值等，在使用之前需要进行数据预处理。数据预处理没有标准的流程，通常针对不同的任务和数据集属性的不同而不同。数据预处理的常用流程为：去除唯一属性、处理缺失值、属性编码、数据标准化正则化、特征选择、主成分分析。

```
Market train shape: (4072956, 16)
```

图 3.1 Market 数据大小

在我们的 Market（市场）数据集中，一共有 400 多万条记录，每条记录有 16 个属性（特征），我们要预测的目标变量是 `returnsOpenNextMktres10`（未来 10 天市场残余回报）

属 性	说 明
time	当前时间（在marketdata中，所有行都是在UTC时间22:00）
assetCode	资产的唯一ID
assetName	与一组assetCodes对应的名称。如果相应的assetCode在新闻数据中没有任何行，则这些可能是“未知”。
universe	一个布尔值，表示当天的工具是否包含在评分中。在训练数据时间段之外不提供该值。特定日期的交易范围是可用于交易的一组工具（评分函数不会考虑不在交易领域中的工具）。交易世界每天都在变化。
volume	当天股票交易量
close	当天收盘价（未调整分割或股息）
open	当天的未平仓价格（未根据拆分或分红进行调整）
returnsClosePrevRaw1	回报
returnsOpenPrevRaw1	marketdata包含通过不同时间跨度计算的各种回报。这组marketdata中的所有回报都具有以下属性：
returnsClosePrevMktres1	(1) 回报总是计算为未平仓（从一个交易日的开盘时间到另一个交易日的开盘时间）或接近收盘价（从一个交易日的收盘时间到另一个交易日的开盘时间）。
returnsOpenPrevMktres1	(2) 回报是原始的，意味着数据不是根据任何基准进行调整，也不是市场残差（Mktres），这意味着整个市场的变动已被考虑，只留下工具固有的变动。
returnsClosePrevRaw10	(3) 可以在任意间隔内计算返回值。这里提供1天和10天的视野。如果它们向后看，则返回标记为'Prev'，如果向前看，则返回标记为'Next'。）
returnsOpenPrevMktres10	
returnsOpenNextMktres10【目标变量】	未来10天，市场残余回报。这是竞争评分中使用的目标变量。市场数据已经过滤，因此returnsOpenNextMktres10始终不为空。

图 3.2 Market 数据集属性介绍

同时市场数据中，存在两种变量，一种是分类变量，一种是数值型变量。在数据预测之前，我们需要分别对这两种数据进行预处理操作。

```
cat_cols = ['assetCode']
num_cols = ['volume', 'close', 'open', 'returnsClosePrevRaw1', 'returnsOpenPrevRaw1', 'returnsClosePrevMktres1',
            'returnsOpenPrevMktres1', 'returnsClosePrevRaw10', 'returnsOpenPrevRaw10', 'returnsClosePrevMktres10',
            'returnsOpenPrevMktres10']
```

图 3.3 不同数据类型的特征

### 3.1.1 分类变量预处理

统计学中的变量（variables）大致可以分为数值变量（numerical）和分类变量（categorical）。数值型变量是值可以取一些列的数，这些值对于 加法、减法、求平均值等操作是有意义的。而分类变量对于上述的操作是没有意义的。

分类变量大体上又可以分为以下两类：一类是有序分类变量（ordinal）：描述事物等级或顺序，变量值可以是数值型或字符型，可以进而比较优劣，如喜欢的程度：很喜欢、一般、不喜欢。另一类是无序分类变量（nominal）：取值之间没有顺序差别，仅做分类，又可分为二分类变量和多分类变量。二分类变量是指将全部数据分成两个类别，如男、女，对、错，阴、阳等，二分类变量是一种特殊的分类变量，有其特有的分析方法。多分类变量是指两个以上类别，如血型分为 A、B、AB、O。

在我们的 Market 数据中，分类变量主要为 assetCode（资产代码），它是由一系列字符串组成，如下图所示（例如纵坐标：BAC.N、GE.N、F.N 等等）。

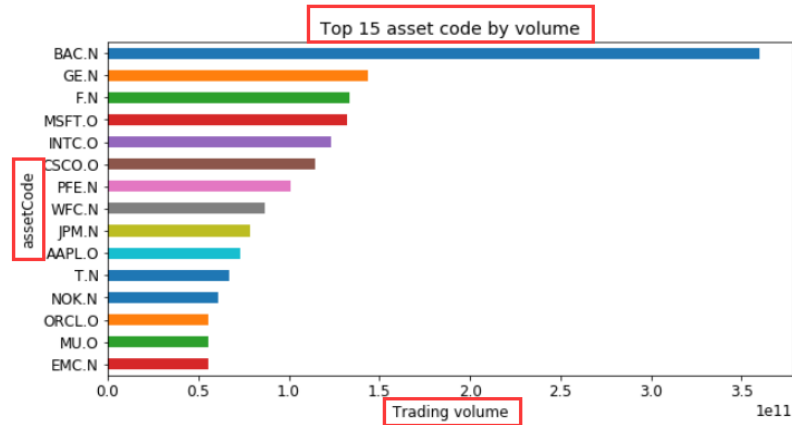


图 3.4 部分资产交易量

传统的机器学习方法一般无法直接处理这类变量。因此，我们需要对其进行编码。对于分类型数据的编码，我们通常会使用两种方式来实现，分别是：`one-hot encoding` 和 `label-encoding`。

```
There are 3780 unique asset code
```

图 3.5 Market 数据中不同资产代码数

虽然 `one-hot encoding` 编码的优点是解决了分类器不好处理分类数据的问题，在一定程度上也起到了扩充特征的作用。它的值只有 0 和 1，不同的类型存储在垂直的空间。但当类别的数量很多时，特征空间会变得非常大，容易造成维度灾难。事实证明，我们的 `AssetCode` 一共有 3700+ 个不同的类别，采用 `one-hot encoding` 编码的话，维度将会相当大，同时也大大增加了计算开销。

因此，我们选择了 `label-encoding` 的编码方式。我们可以自定义量化数字，因为资产代码本身相对比较独立，且只是一个用来区别的符号，并不需要太强的解释性。所以 `label-encoding` 编码可解释性差的缺点，对于我们的数据并不存在。

### 3.1.2 数值型变量预处理

数值变量又可以分为两类：离散型变量(`discrete`)和连续型变量(`continuous`)。离散型变量 (`discrete`)：值只能用自然数或整数单位计算，其数值是间断的，相邻两个数值之间不再有其他数值，这种变量的取值一般使用计数方法取得。连续型变量 (`continuous`)：在一定区间内可以任意取值，其数值是连续不断的，相邻两个数值可作无限分割，即可取无限个数值。如身高、绳子的长度等。

数值型变量的处理方法一般有异常值检测、缺失值处理、标准化操作等。

### (一) 异常值检测、处理方面

经过大量调研,我们发现美股普通类股票价格无涨跌幅限制,但实行熔断机制。根据该机制,价格不低于 1 美元的股票或交易所交易基金(ETF)在 5 分钟内波幅达到或超过 30%,将被暂停交易;价格低于 1 美元的证券在 5 分钟内波幅达到或超过 50%,将被暂停交易。

因此按照熔断机制,我们设定收盘价(close)和开盘价(open)的差值不能超过 $\pm 50\%$ ,超过即视为异常值,滤除超过 50%的异常记录。

```
In 83 lines price increases by 50% or more in a day
In 16 lines price decreases by 50% or more in a day
```

图 3.6 异常值数据统计

### (二) 缺失值检测、处理方面:

首先,我们统计了每一个属性缺失值个数情况,如下图所示。

```
time          0
assetCode     0
assetName     0
volume        0
close         0
open          0
returnsClosePrevRaw1  0
returnsOpenPrevRaw1  0
returnsClosePrevMktres1  15980
returnsOpenPrevMktres1  15988
returnsClosePrevRaw10  0
returnsOpenPrevRaw10  0
returnsClosePrevMktres10  93810
returnsOpenPrevMktres10  93054
returnsOpenNextMktres10  0
universe      0
dtype: int64
```

图 3.7 缺失值数据统计

缺失出现在市场调整回报数据中,我们用原始回报数据(raw return)填充对应有的缺失值。

### (三) 标准化操作(去均值和方差进行缩放)

数据标准化:当单个特征的样本取值相差甚大或明显不遵从高斯正态分布时,标准化表现的效果较差。实际操作中,经常忽略特征数据的分布形状,移除每个特征均值,划分离散特征的标准差,从而等级化,进而实现数据中心化。



为了避免某一属性值过大对我们的预测结果产生影响，我们进行了标准化处理。将所有属性的数据值转化成均值：0，标准差：1 的形式。提升模型的收敛速度，提升模型的精度。

具体算法流程如下：

$$\text{均值公式: } \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i$$

$$\text{方差公式: } s^2 = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2$$

标准化公式为： $\text{scaler} = (X - X\_mean) / X\_std$  计算时对每个属性/每列分别进行。本次处理中，采用 `sklearn.preprocessing.StandardScaler` 类对数值型数据进行标准化处理。

### 3.2 网络结构

本次比赛中，所有的代码均在 Kaggle 平台上编写。尽管平台支持 GPU 计算，但用 GPU 计算出来的结果无法提交。鉴于此，深度学习等复杂神经网络模型不适用于此比赛，需要构造一些精巧、浅层神经网络模型来进行预测。如下图所示。

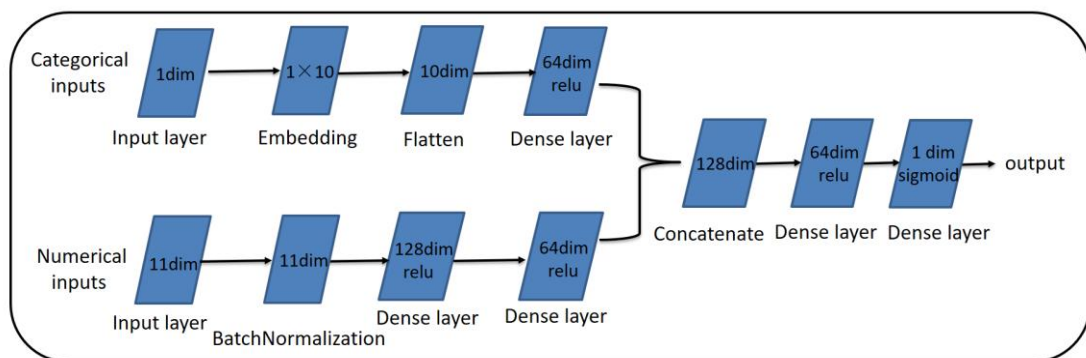


图 3.8 神经网络模型框架

整个神经网络基于 Keras 库进行编写。输入部分由两部分组成，分别为分类变量和数值型变量。对于分类变量，我们首先经过一个 Embedding（嵌入层），将正整数转换为具有固定大小的向量，如[[4],[20]]->[[0.25,0.1],[0.6,-0.2]]；再经过 Flatten 层用来将输入“压平”，把多维的输入一维化，Flatten 不影响 batch 的大小，如将[[1,2,3,4,5,6,7,8,9,10]]转化为[1,2,3,4,5,6,7,8,9,10]；最后将 Flatten 层的输出传

入到一个 64 维输出的全连接层当中。

对于数值型变量, 经过预处理后, 先经过 BatchNormalization 层, 在每个 batch 上将前一层的激活值重新规范化, 使得其输出数据的均值接近 0, 其标准差接近 1; 再将 BatchNormalization 层的输出一次传入到一个 128 维、64 维输出的全连接层当中。

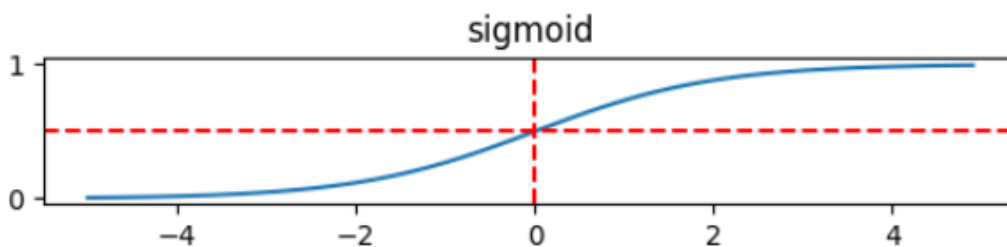
Concatenate 层用于将分类和数值型特征进行联合, 作为总体输入 (128 维), 经过 64 维全连接层, 最后经过一个全连接层得到我们的 1 维的 output (输出)。

```
# Lets print our model
model.summary()
```

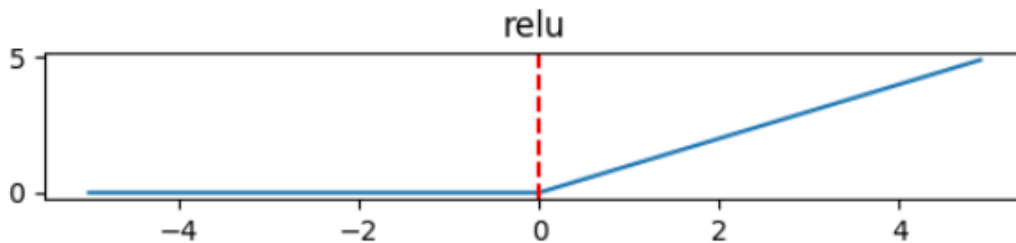
Layer (type)	Output Shape	Param #	Connected to
num (InputLayer)	(None, 11)	0	
assetCode (InputLayer)	(None, 1)	0	
batch_normalization_2 (BatchNor	(None, 11)	44	num[0][0]
embedding_2 (Embedding)	(None, 1, 10)	37810	assetCode[0][0]
dense_7 (Dense)	(None, 128)	1536	batch_normalization_2[0][0]
flatten_2 (Flatten)	(None, 10)	0	embedding_2[0][0]
dense_8 (Dense)	(None, 64)	8256	dense_7[0][0]
dense_6 (Dense)	(None, 64)	704	flatten_2[0][0]
concatenate_2 (Concatenate)	(None, 128)	0	dense_8[0][0] dense_6[0][0]
dense_9 (Dense)	(None, 64)	8256	concatenate_2[0][0]
dense_10 (Dense)	(None, 1)	65	dense_9[0][0]

图 3.9 模型中连接关系及每层 shape

上图打印出了每层神经网络的信息, 可以得到连接关系以及每一层的 shape。



$$\text{sigmoid: } y = 1/(1 + e^{-x})$$



$$\text{relu: } y = \max(0, x)$$

图 3.10 sigmoid 与 relu 激活函数

在激励函数方面，除了最后一层选择 sigmoid，其余各层均选用 relu 作为我们的激励函数。

优化器方面，选择 Adam 作为我们的优化器。

```
model = Model(inputs = categorical_inputs + [numerical_inputs], outputs=out)
model.compile(optimizer='adam', loss=binary_crossentropy)
```

图 3.11 优化器与 loss 函数

loss 函数方面，因为我们输出层选择 sigmoid 作为激励函数，在训练神经网络过程中，我们通过梯度下降算法来更新  $w$  和  $b$ 。如果选择方差代价函数（即采用均方误差 MSE）作为我们的 loss 函数，因为 sigmoid 函数的性质，会导致  $\sigma'(z)$  在  $z$  取大部分值时会很小，这样会使得  $w$  和  $b$  更新非常慢（因为  $\eta * a * \sigma'(z)$  这一项接近于 0）。Binary\_crossentropy（交叉熵代价函数），对  $w$  和  $b$  的导数中没有  $\sigma'(z)$  这一项，权重的更新是受  $\sigma(z) - y$  这一项影响，即受误差的影响。所以当误差大的时候，权重更新就快，当误差小的时候，权重的更新就慢。

$$C = -\frac{1}{n} \sum_x [y \ln a + (1 - y) \ln(1 - a)]$$

$$\frac{\partial C}{\partial b} = \frac{1}{n} \sum_x (\sigma(z) - y)$$

参数情况如下图所示：

```
Total params: 56,671
Trainable params: 56,649
Non-trainable params: 22
```

图 3.12 参数情况

### 3.3 训练与预测过程

```
Train on 3665660 samples, validate on 407296 samples
Epoch 1/3
3665660/3665660 [=====] - 267s 73us/step - loss: 0.6826 - val_loss:
0.6831

Epoch 0001: val_loss improved from inf to 0.68306, saving model to model.hdf5
Epoch 2/3
3665660/3665660 [=====] - 266s 73us/step - loss: 0.6821 - val_loss:
0.6849

Epoch 0002: val_loss did not improve from 0.68306
Epoch 3/3
3665660/3665660 [=====] - 265s 72us/step - loss: 0.6816 - val_loss:
0.6823

Epoch 0003: val_loss improved from 0.68306 to 0.68228, saving model to model.hdf5
```

图 3.13 训练过程

Market 数据总共有四百多万条，我们取出 1/10 的数据集用来验证，训练当中，耗时 3 个 epoch，总时长约 13 分钟。预测结果中，置信度方面，验证集和测试集的置信度分布相似。

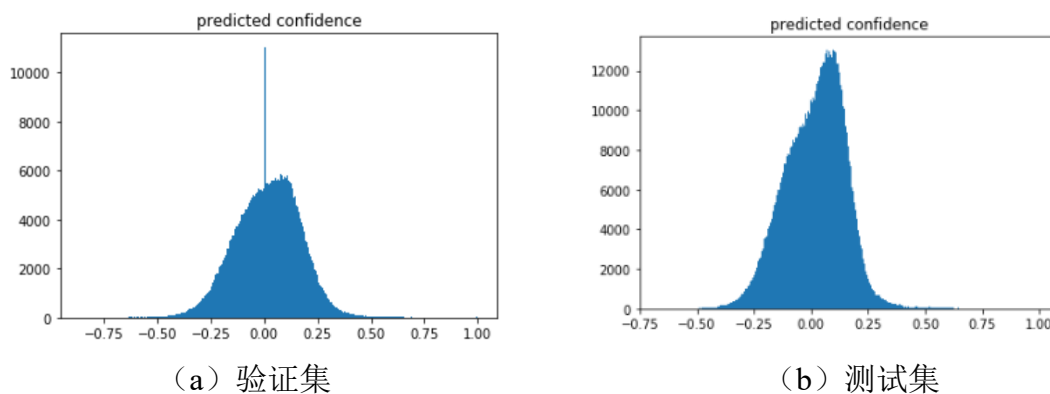


图 3.14 验证集与测试集置信度分布图

### 3.4 排名情况

神经网络模型，当时排名：TOP 12%

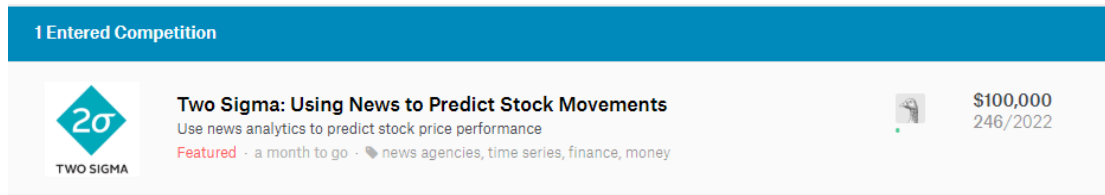


图 3.15 神经网络模型排名情况

神经网络模型是模型尝试阶段使用的模型，使用 Market 数据进行预测，到了后期，该模型对预测结果的提升效果不太明显。所以，比赛后期，我们采用 lightGBM 模型。

## 4 Boosting 方法-LightGBM 模型

### 4.1 Market 数据特征工程

数据的特征工程与预处理环节与神经网络模型的前期工作大部分一致，但后来发现，由于使用原始特征样本直接预测，预测结果波动性过大，加入数据局部特征，可减小预测结果波动性。

#### 原始特征：

考虑到验证集数据没有'returnsOpenNextMktres10'、'universe'这两项特征，因此在此在 Market 原始特征选择上，将这两项特征删除。具体特征如下：

```
['time', 'assetCode', 'assetName', 'volume', 'close', 'open',  
 'returnsClosePrevRaw1', 'returnsOpenPrevRaw1',  
 'returnsClosePrevMktres1', 'returnsOpenPrevMktres1',  
 'returnsClosePrevRaw10', 'returnsOpenPrevRaw10',  
 'returnsClosePrevMktres10', 'returnsOpenPrevMktres10']
```

#### 添加特征：

由于只使用原始 Market 特征进行预测，会导致预测结果波动性过大，因此这里加入数据局部特征，减小了预测结果波动性。

这里，取'returnsClosePrevMktres10'、'returnsClosePrevRaw10'、'open'、'close'四个特征分别进行 5 天、10 天、15 天、20 天的平均数(mean)、最大值(max)、最

小值(min)计算，作为 Market 数据的局部特征。具体特征如下：

```
['returnsClosePrevMktres10_lag_5_mean',  
 'returnsClosePrevMktres10_lag_5_max',  
 'returnsClosePrevMktres10_lag_5_min',  
 'returnsClosePrevMktres10_lag_10_mean',  
 'returnsClosePrevMktres10_lag_10_max',  
 'returnsClosePrevMktres10_lag_10_min',  
 'returnsClosePrevMktres10_lag_15_mean',  
 'returnsClosePrevMktres10_lag_15_max',  
 'returnsClosePrevMktres10_lag_15_min',  
 'returnsClosePrevMktres10_lag_20_mean',  
 'returnsClosePrevMktres10_lag_20_max',  
 'returnsClosePrevMktres10_lag_20_min',  
 'returnsClosePrevRaw10_lag_5_mean', 'returnsClosePrevRaw10_lag_5_max',  
 'returnsClosePrevRaw10_lag_5_min', 'returnsClosePrevRaw10_lag_10_mean',  
 'returnsClosePrevRaw10_lag_10_max', 'returnsClosePrevRaw10_lag_10_min',  
 'returnsClosePrevRaw10_lag_15_mean', 'returnsClosePrevRaw10_lag_15_max',  
 'returnsClosePrevRaw10_lag_15_min', 'returnsClosePrevRaw10_lag_20_mean',  
 'returnsClosePrevRaw10_lag_20_max', 'returnsClosePrevRaw10_lag_20_min',  
 'open_lag_5_mean', 'open_lag_5_max', 'open_lag_5_min',  
 'open_lag_10_mean', 'open_lag_10_max', 'open_lag_10_min',  
 'open_lag_15_mean', 'open_lag_15_max', 'open_lag_15_min',  
 'open_lag_20_mean', 'open_lag_20_max', 'open_lag_20_min',  
 'close_lag_5_mean', 'close_lag_5_max', 'close_lag_5_min',  
 'close_lag_10_mean', 'close_lag_10_max', 'close_lag_10_min',  
 'close_lag_15_mean', 'close_lag_15_max', 'close_lag_15_min',  
 'close_lag_20_mean', 'close_lag_20_max', 'close_lag_20_min']
```

从图 4.1 和图 4.2 可以清晰地看出，Market 数据局部特征对于预测结果的影响。

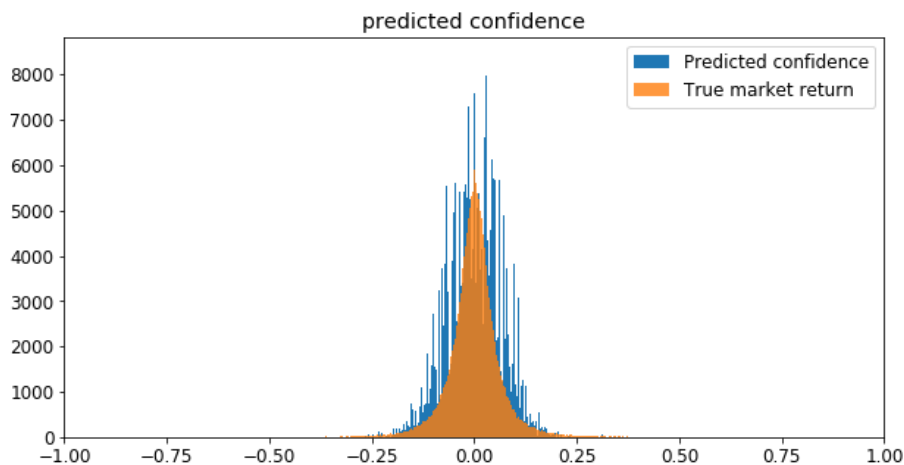


图 4.1 未添加 Market 数据局部特征的预测结果

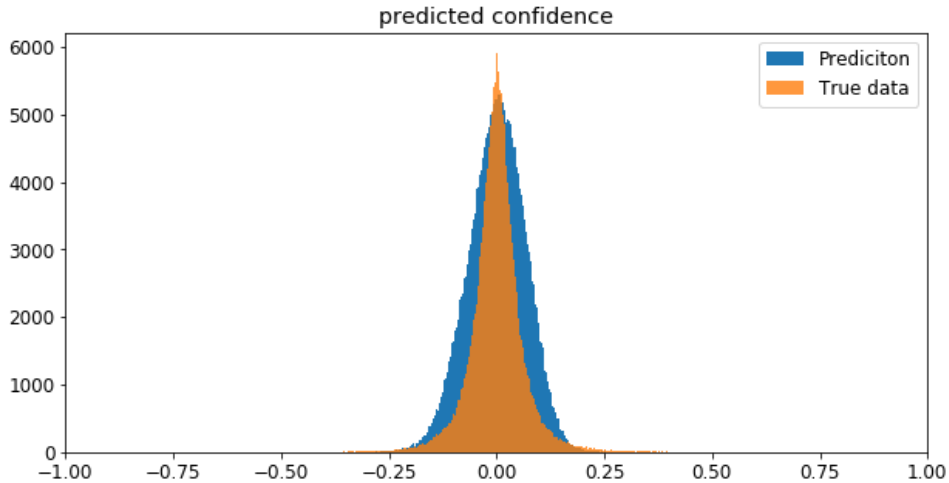


图 4.2 添加 Market 数据局部特征后的预测结果

## 4.2 News 数据选取

由于 LightGBM 模块属于基于树的模型，能够很好地展现出特征对决策地影响，在先前的尝试中，我们通过 Feature Importance 模块评估了特征重要性。

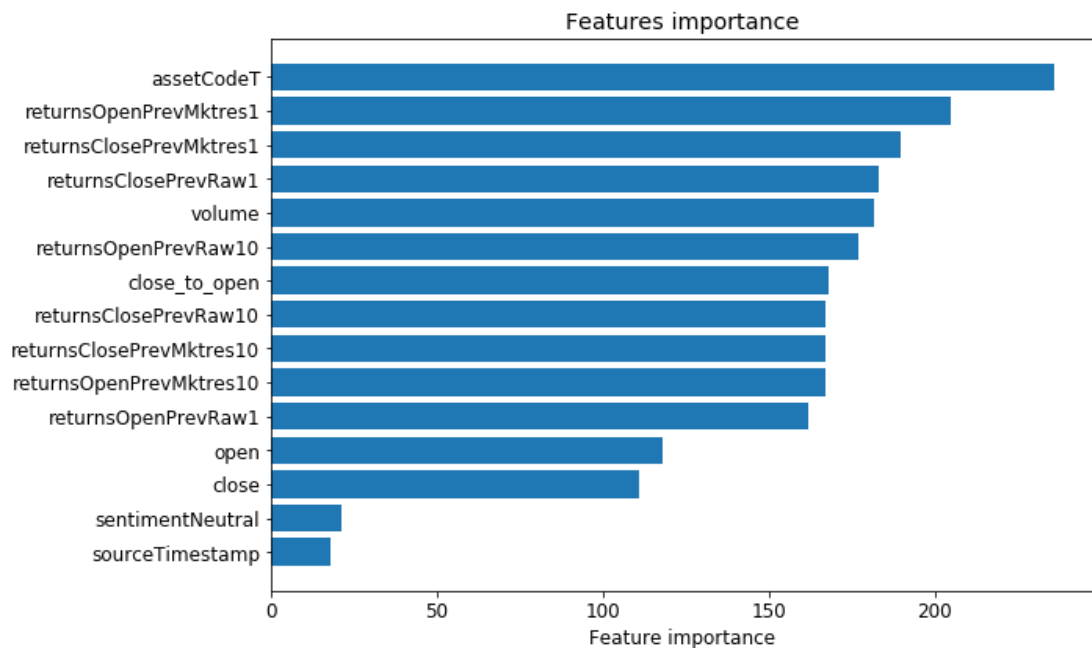


图 4.3 特征重要性评估

可以看出，新闻数据只有 sentiment 对结果有一定的影响。所以，我们从 News 数据中选取 sentimentNegative、sentimentNeutral、sentimentPositive 三个特征作为样本特征。该特征反映了该条新闻对资产的情感倾向，在一定程度上会影响下一



日公司股票的走势。

### 4.3 特征融合

#### 时间选择:

如神经网络模型中所述,2008 金融危机,以及 2009 年过渡期数据的不稳定,我们在市场数据和新闻数据方面均选择了 2010 年 1 月 1 日至今的数据。

#### 填充空数据:

当数据类型为“object”,用“other”填充;当数据类型为“int64”或“float64”,用“平均数”填充。

接着,以原始市场数据为母版,采用 `pd.merge` 方法将所添加的市场局部特征和新闻特征依次融合。值得注意的是,市场数据和新闻数据并非一一对应关系,而是多对多的交叉对应关系。经过前期数据分析,我们得出“市场数据对于预测准确度影响更加显著”的结论。因此,这里在市场数据和新闻数据融合过程中,我们以市场数据为基础进行新闻特征拓展,舍弃了部分没有市场数据对应的新闻数据。

具体数据融合代码如下:

```
market_train_df = pd.merge(market_train_df,new_df,how='left',on=['time',
'assetCode']) # 原始 market 数据融合新添加的 market 数据
market_train_df = pd.merge(market_train_df,news_train_df,how='left',on=
['time','assetName']) # market 数据融合 news 数据
```

经过融合后的总体特征为:

```
['time', 'assetCode', 'assetName', 'volume', 'close', 'open',
'returnsClosePrevRaw1', 'returnsOpenPrevRaw1',
'returnsClosePrevMktres1', 'returnsOpenPrevMktres1',
'returnsClosePrevRaw10', 'returnsOpenPrevRaw10',
'returnsClosePrevMktres10', 'returnsOpenPrevMktres10',
'returnsClosePrevMktres10_lag_5_mean',
'returnsClosePrevMktres10_lag_5_max',
'returnsClosePrevMktres10_lag_5_min',
'returnsClosePrevMktres10_lag_10_mean',
'returnsClosePrevMktres10_lag_10_max',
'returnsClosePrevMktres10_lag_10_min',
'returnsClosePrevMktres10_lag_15_mean',
```

```
'returnsClosePrevMktres10_lag_15_max',  
'returnsClosePrevMktres10_lag_15_min',  
'returnsClosePrevMktres10_lag_20_mean',  
'returnsClosePrevMktres10_lag_20_max',  
'returnsClosePrevMktres10_lag_20_min',  
'returnsClosePrevRaw10_lag_5_mean', 'returnsClosePrevRaw10_lag_5_max',  
'returnsClosePrevRaw10_lag_5_min', 'returnsClosePrevRaw10_lag_10_mean',  
'returnsClosePrevRaw10_lag_10_max', 'returnsClosePrevRaw10_lag_10_min',  
'returnsClosePrevRaw10_lag_15_mean', 'returnsClosePrevRaw10_lag_15_max',  
'returnsClosePrevRaw10_lag_15_min', 'returnsClosePrevRaw10_lag_20_mean',  
'returnsClosePrevRaw10_lag_20_max', 'returnsClosePrevRaw10_lag_20_min',  
'open_lag_5_mean', 'open_lag_5_max', 'open_lag_5_min',  
'open_lag_10_mean', 'open_lag_10_max', 'open_lag_10_min',  
'open_lag_15_mean', 'open_lag_15_max', 'open_lag_15_min',  
'open_lag_20_mean', 'open_lag_20_max', 'open_lag_20_min',  
'close_lag_5_mean', 'close_lag_5_max', 'close_lag_5_min',  
'close_lag_10_mean', 'close_lag_10_max', 'close_lag_10_min',  
'close_lag_15_mean', 'close_lag_15_max', 'close_lag_15_min',  
'close_lag_20_mean', 'close_lag_20_max', 'close_lag_20_min',  
'sentimentNegative', 'sentimentNeutral', 'sentimentPositive']
```

## 4.4 LightGBM 模型

### 4.4.1 模型介绍

LightGBM 是微软推出了一个轻量级的 boosting 框架，具有快的训练效率、低内存使用、高的准确率等特点。因为此次比赛只可以运行在服务器段，并且不可以使用 GPU 加速，LightGBM 十分适合此次比赛。

LightGBM 主要有以下特点：基于 Histogram 的决策树算法、带深度限制的 Leaf-wise 的叶子生长策略。其中，Histogram 直方图算法的基本思想是：先把连续的浮点特征值离散化成 k 个整数，同时构造一个宽度为 k 的直方图。遍历数据时，根据离散化后的值作为索引在直方图中累积统计量，当遍历一次数据后，直方图累积了需要的统计量，然后根据直方图的离散值，遍历寻找最优的分割点。决策树中 Level-wise 算法过一次数据可以同时分裂同一层的叶子，容易进行多线程优化，也好控制模型复杂度，不容易过拟合。但实际上很多叶子的分裂增益较低，没必要进行搜索和分裂，而 Level-wise 算法不加区分的对待同一层的叶子，

就带来了许多不必要的开销，导致计算效率的低下。Leaf-wise 则是一种更为高效的策略：每次从当前所有叶子中，找到分裂增益最大的一个叶子，然后分裂，如此循环。同 Level-wise 相比，在分裂次数相同的情况下，Leaf-wise 可以降低更多的误差，得到更好的精度。同时，为避免 Leaf-wise 算法可能长出较深的决策树，产生过拟合，LightGBM 在 Leaf-wise 之上增加了一个最大深度限制，在保证高效率的同时防止过拟合。LightGBM 模型框架如下图。

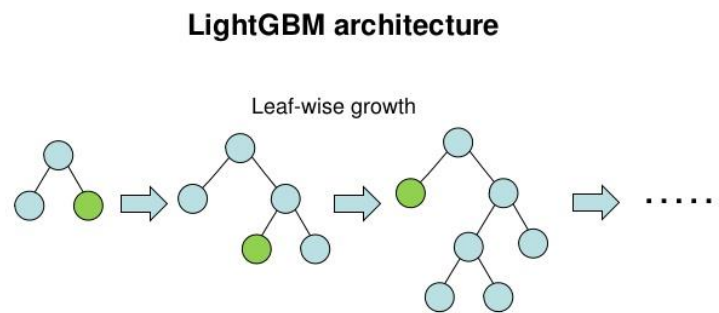


图 4.4 LightGBM 框架

#### 4.4.2 参数选择

我们利用 GridSearchCV 和 RandomizedSearchCV 方法各寻了几次最优参数，最终选择了如下两组参数分别训练两个 LightGBM 模型，并将最后的结果取平均作为最终 confidence 指标。

```
params_1 = {  
    'task': 'train',  
    'boosting_type': 'gbdt',  
    'objective': 'binary',  
    'learning_rate': 0.19000424246380565,  
    'num_leaves': 2452,  
    'min_data_in_leaf': 212,  
    'num_iteration': 239,  
    'max_bin': 202,  
    'verbose': 1  
}
```

```
params_2 = {  
    'task': 'train',  
    'boosting_type': 'gbdt',  
    'objective': 'binary',  
    'learning_rate': 0.19016805202090095,
```

```
'num_leaves': 2583,  
'min_data_in_leaf': 213,  
'num_iteration': 172,  
'max_bin': 220,  
'verbose': 1  
}
```

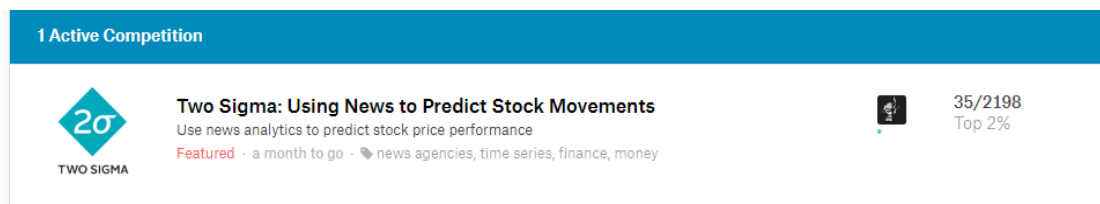
其中，模型训练过程中，X 变量为之前选取及建立的最终特征；Y 变量表示当前股票是否上涨。

可以看到，我们选取的任务目标 `object` 不是 `regression`，而是用了 `binary`，即我们把这个任务定为了二分类模型。原因主要是，由于我们所要预测的收益率过于接近于 0，数据间差别较小，若定义为回归任务，结果会非常差。

所以，我们将任务定为 `Binary`，即把 0 作为负回报，1 作为正回报，体现在若 `returnsOpenNextMktres10` 大于 0，记 Y 为 1；若 `returnsOpenNextMktres10` 小于 0，记 Y 为 0。`LightGBM` 可以使用 `predict_proba` 计算出属于两个类别的概率，简明来说就是，股票趋近于正回报结果的概率，即是我们的预期收益率。

## 4.5 排名情况

在特征工程做了大量工作后，我们用 `LightGBM` 在 12 月底跑出了测试集上分数 0.95，当时提交到 `Leaderboard` 的分数为 0.72210（Top 2%）。



1 Active Competition

**Two Sigma: Using News to Predict Stock Movements**  
Use news analytics to predict stock price performance  
Featured · a month to go · news agencies, time series, finance, money

35/2198  
Top 2%

## 5 团队分工与最终成绩

团队分工：

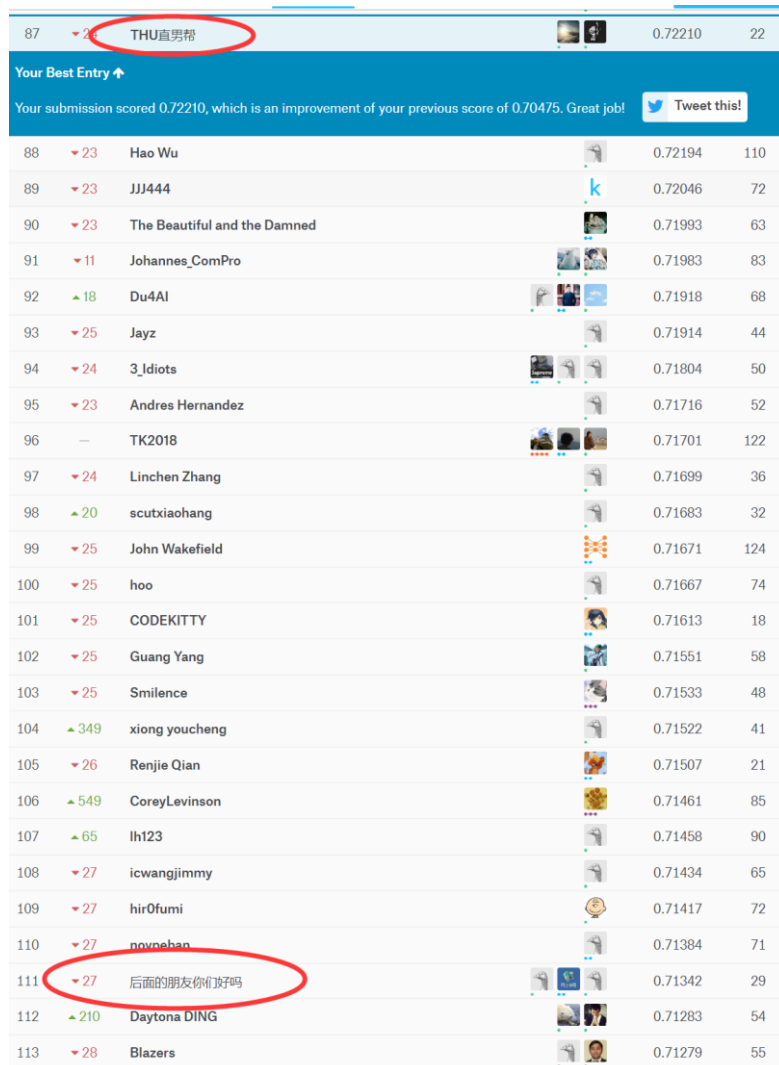
表 5.1 团队分工表

团队成员	负责工作
------	------

徐哲、张凌霄、路万通	建立模型(NN 及 boosting 方法)、特征工程、编写代码与调参
许卓群、于苗苗	项目背景调研、编程对数据进行 EDA 分析及总结

### 最终成绩:

该比赛于 2019 年 1 月 8 日结束，很多高手后面发力，我们小组“**THU 直男帮**”在 Leaderboard 上排名有一定下滑，最终为：**Top4%**，共有 2800+的参赛队伍。另一门《统计机器学习》课程的小组“后面的朋友你们好吗”同样也是做这个比赛，我们的排名略高于他们。



Rank	Change	Team Name	Score	Submissions
87	▼2	THU直男帮	0.72210	22
<b>Your Best Entry</b>				
Your submission scored 0.72210, which is an improvement of your previous score of 0.70475. Great job!				
88	▼23	Hao Wu	0.72194	110
89	▼23	JJJ444	0.72046	72
90	▼23	The Beautiful and the Damned	0.71993	63
91	▼11	Johannes_ComPro	0.71983	83
92	▲18	Du4AI	0.71918	68
93	▼25	Jayz	0.71914	44
94	▼24	3 Idiots	0.71804	50
95	▼23	Andres Hernandez	0.71716	52
96	—	TK2018	0.71701	122
97	▼24	Linchen Zhang	0.71699	36
98	▲20	scutxiaohang	0.71683	32
99	▼25	John Wakefield	0.71671	124
100	▼25	hoo	0.71667	74
101	▼25	CODEKITTY	0.71613	18
102	▼25	Guang Yang	0.71551	58
103	▼25	Smilence	0.71533	48
104	▲349	xiong youcheng	0.71522	41
105	▼26	Renjie Qian	0.71507	21
106	▲549	CoreyLevinson	0.71461	85
107	▲65	lh123	0.71458	90
108	▼27	icwangjimmy	0.71434	65
109	▼27	hir0fumi	0.71417	72
110	▼27	novneban	0.71384	71
111	▼27	后面的朋友你们好吗	0.71342	29
112	▲210	Daytona DING	0.71283	54
113	▼28	Blazers	0.71279	55

图 5.1 Leaderboard 情况

## 6 个人感想

在最初确定选题方面，我们组就考虑到想做一个要结合金融领域知识的比赛。在比赛过程中，面临的最大问题其实是计算资源的短缺，因为由于主办方比较重视数据保护，因此要求程序只可以在服务器上面运行，并且没有 GPU 加速。这就给我们的模型和数据特征的选择提出了更高要求。

如今回望这次比赛以及课程的学习，在人工智能与数据挖掘领域，有一个问题值得我们反思：现在的深度学习浪潮，太过于强调黑箱理论，反而逐渐忽视了特征选择这一传统机器学习的基本步骤。现在的产业界强调领域知识与机器学习的结合，因此如何更好的利用数据，面对不同场景选择适当的特征，提高算法的运行效率，也是人工智能很重要的一个发展方向。

通过比赛，我也学习了之前所没有接触过的模型。其实，很多结果优秀的模型都是根据基本的模型改进而来。因此，在今后的科研和工作当中，熟练掌握基本模型，并发现其中改进可能，最终利用数学的手段进行理论推导，也是我们应有的技能。

最后，感谢课程中老师对我们的指导，感谢我的团队成员的支持。